APPENDIX: FEHM VERIFICATION SCRIPTS

A. DESCRIPTION OF SCRIPTS

A series of *cshell* scripts were developed to perform the verification operations (see Table X). A primary script, **F**EHM **V**&**V** **S**cript for **E**xecution of **C**omparison **T**ests (**FEHM_VVSECT**), controls the FEHM verification runs via an execution script (**RUN_VERIFICATION**) then generates a comparison of results and summary via a supporting comparison tests script (**COMPARE_RESULTS**). An execution log is generated when the primary script is run (see Sections E through G for listings of the three top-level scripts and Sections H and I for examples of the specific problem execution and run comparison scripts). In addition to being invoked by the **FEHM_VVSECT** script, the **RUN_VERIFICATION** and **COMPARE_RESULTS** scripts may be invoked independently to re-execute specific tests or comparisons by providing problem descriptors as arguments. Two subsidiary programs, **COMPARE** and **SUMMARIZE**, used by the comparison tests script, are also described in Table X.

Table X. Scripts and support programs for verification operations

Script name	Description
FEHM_VVSECT	Executes all verification test problems (**RUN_VERIFICATION**) and generates comparison of results and summary (**COMPARE_RESULTS**) usage: **FEHM_VVSECT** [xfehmn_path verification_dir]
RUN_VERIFICATION	Invokes script for each verification test problem (**run*problem***) usage: **RUN_VERIFICATION** [-p xfehmn_path verification_dir] [*problem1 problem2* . . .]
COMPARE_RESULTS	Invokes script to generate result comparison and summary for each verification test problem (**comp*problem***) usage: **COMPARE_RESULTS** [-p verification_dir] [-i date_id] [*problem1 problem2* . . .]
run*problem*	Problem-specific script for **FEHM** input/output setup and code execution
comp*problem*	Problem-specific script for **COMPARE** and **SUMMARIZE** input/ output setup and execution
MAKE_OUTPUT_DIRS	Script for making output directories for verification problems in a user's local verification directory
COMPARE	Program that reads **FEHM** results and generates a numeric comparison with analytical or alternate code solutions.
SUMMARIZE	Program that reads results from **COMPARE** and outputs the results for related groups of tests in a single table.

The problems are set up in a directory-tree file structure. The root/verification directory contains the primary (**FEHM_VVSECT**), execution (**RUN_VERIFICATION**), and comparison (**COMPARE_RESULTS**) scripts with a subdirectory for each test problem. The execution log and summary report are written to the root directory (see Sections J and K). Each problem directory contains the problem-specific execution (**run*problem***)

and comparison (**comp*problem***) scripts, an input and output subdirectory, and other files and directories as needed (Fig. 19). The following problems are currently run by the **FEHM_VVSECT** script (the section numbers in parentheses correspond to the problem and result descriptions found in Dash et al. 1997): avdonin (5.8), dissolution (5.15), doe5a (5.10), dryout (5.11), dual (5.7), fracture_transport (5.14), heat2d (5.2), heat3d (5.2), henrys_law (5.13), infiltration (5.5), multi_solute (5.16), ramey (5.3), sorption (5.12), theis (5.4), toronyi (5.9), transport3D (5.17), and vapor_extraction (5.6). Scripts for this problem are being developed so it may be run independently. As additional test problems are developed, they will be incorporated into the test environment. The thermodynamics tests (5.1) are run independently because the functions need only be retested if the polynomial coefficients are modified. Also, any errors introduced to the routines containing the thermodynamic functions would result in errors in the other tests.

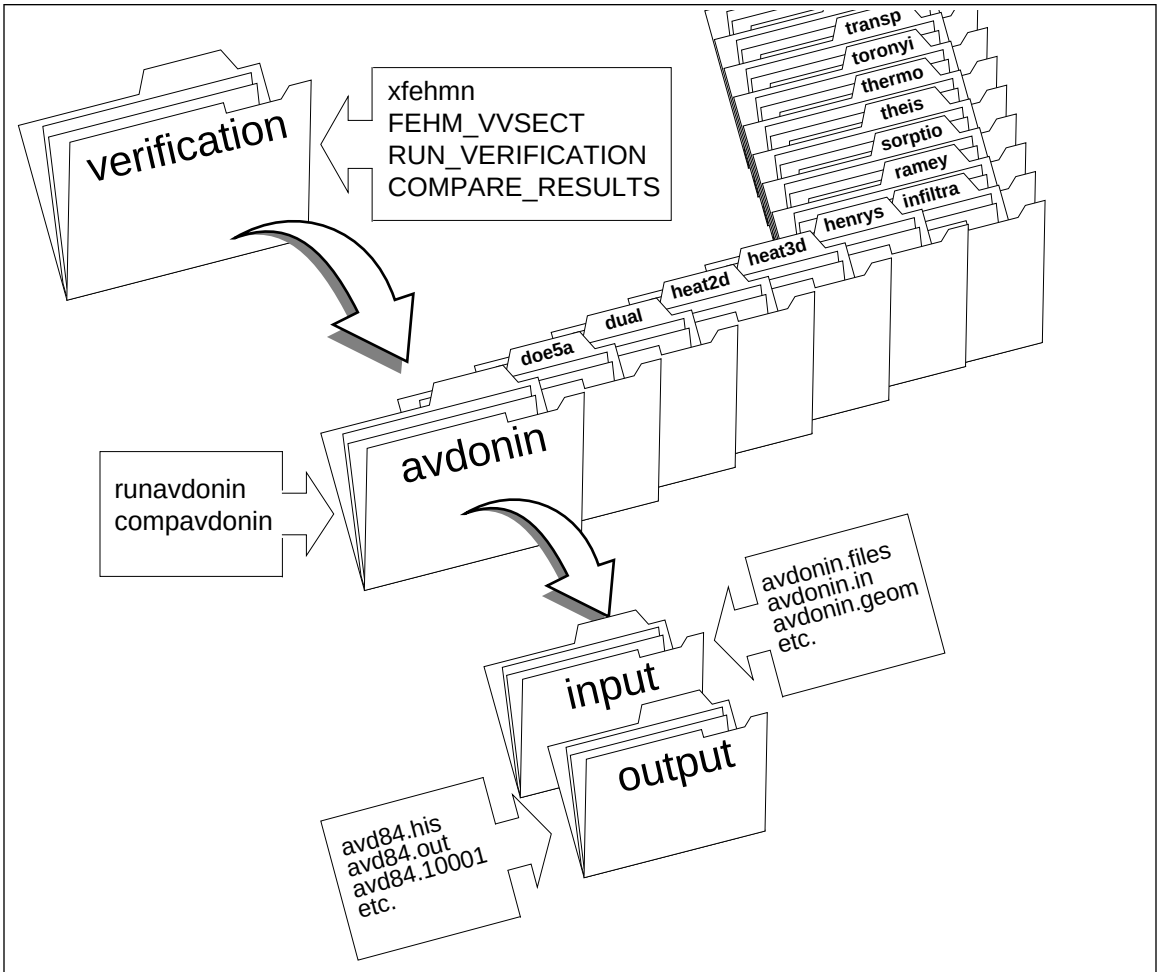

Figure 19. Diagram of verification directory structure

B. INSTALLATION

Files needed to set up the verification environment are contained in a tar file called verification.tar. A directory should be created and the files installed there, i.e.,

```
% mkdir verification
```

% cd verification

% tar xvf *directory*/verification.tar

where *directory* is the location of the verification.tar file.

When the files have been installed in the verification directory, the auxiliary programs need to be built and installed. To do this, change into verification subdirectory SRC. Currently, there are two subdirectories under SRC: compare and thermodynamics. The compare subdirectory contains programs (COMPARE, COMPARET, and SUMMARIZE) that are used to compare FEHM results to analytical and other model solutions and to summarize the comparisons. The thermodynamics subdirectory contains programs (COMPSAT and COMPTHER) that use the FEHM thermodynamic functions to generate values over the valid range of use. The programs require that the SRCDIR environment variable in each Makefile be set to the directory that contains the FEHM source code so that source for the mallocf and parser routines may be found. In addition, program COMPSAT uses source code for the psat function and common files comai.h and comdti.h.

To build programs COMPARE, COMPARET, and SUMMARIZE:

% cd SRC/compare

% make -f Makefile.compare *machine*

where *machine* is sun, hp, ibm, sgi, or cray.

Links from the thermodynamics subdirectory may be made to comai.h and comdti.h of the FEHM source code to use the most up-to-date versions of these files during compilation or the files may be copied to the thermodynamics subdirectory. Common file comii.h has been modified for use with these test programs so the version in the FEHM objects directory currently cannot be used. To build programs COMPSAT and COMPTHER:

% cd SRC/thermodynamics

% make -f Makefile.thermo all

The verification scripts expect the auxiliary program binaries to reside at the top level of the verification directory. This requirement can be accomplished either by moving the binaries there or by making links to their actual location if they are installed elsewhere.

C. USING THE VERIFICATION SCRIPTS

To run the verification tests the user should change to the directory where the output should reside. (For the owner of the verification directory, this is usually the verification directory.) This directory should contain a subdirectory for each verification problem to be run and each problem subdirectory should contain an output subdirectory. A script called MAKE_OUTPUT_DIRS, located in the verification directory, can be used to create (in the current directory) a directory and output subdirectory for each problem.

The user has two options for defining the location of the verification directory (the directory containing verification scripts, problem input files, etc.) and the FEHM executable. The variables may either be entered as command line arguments to the test scripts (see "usage" in Table X) or they may be defined in a file called PATHS in the directory from which the scripts will be executed (see Fig. 20). Variables defined on the command line have precedence over values set in the PATHS file. Complete path names must be used.

```
#! /bin/csh
#       PATHS

#  Set location where FEHM executable can be found:

setenv XFEHMN  /home/fehm/bin/xfehmn.V01.00

#  Set location of verification directory.
#  Verification scripts and executables should be found here, along with
#  problem subdirectories and their associated scripts and input files, i.e.,
#  VER_DIR/problem_dir/input, output directories may be located here
#  or elsewhere

setenv VER_DIR  /home/fehm/verification
```

Figure 20. Example of PATHS file in script-execution directory

To run all verification test problems use **FEHM_VVSECT**, i.e.,

```
% FEHM_VVSECT (assumes PATHS file is present)
```

or

```
% FEHM_VVSECT -p ~/bin/myxfehmn /home/fehm/verification
```

The execution directory will contain a log of the runs and result comparisons in files called VERIFICATION.*date* and SUMMARY_RPT.*date*, where *date* is the date of execution in yymmdd format (see Sections J and K for an example of these files). It should be noted that if a log file or summary file with the current date or identifier already exists in the execution directory it will be renamed, i.e., VERIFICATION.*date*.old or SUMMARY_RPT.*date*.old.

To run selected problems use **RUN_VERIFICATION** and **COMPARE_RESULTS**, i.e.,

```
% RUN_VERIFICATION thermodynamics

% COMPARE_RESULTS -i thermodynamics thermodynamics
```

or

```
% RUN_VERIFICATION -p ~/bin/myxfehmn /home/fehm/verification doe5a dual

% COMPARE_RESULTS doe5a dual
```

For the first case, the summary is found in SUMMARY_RPT.thermodynamics, whereas for the second case, it is found in SUMMARY_RPT.*date* because an identifier was not specified. The execution sequence is logged to the terminal unless output is redirected when the scripts are invoked.

If **RUN_VERIFICATION** and **COMPARE_RESULTS** are invoked without arguments, all test problems and comparisons will be run.

D. ASSUMPTIONS AND LIMITATIONS

The verification scripts were developed on a Sun-4 architecture and have been tested on HP, IBM, SGI, and Sun, but they should work on any standard UNIX workstation. The examples provided below were run on a Sun.

E. FEHM_VVSECT

FEHM_VVSECT script

```
#! /bin/csh
#       FEHM_VVSECT
#       FEHM V&V Script for Execution of Comparison Tests

set date=`date +%y%m%d`
set results=VERIFICATION.$date

if (-f $results) then
        mv $results $results.old
endif

@ paths = 0
# If the PATHS are present as command line arguments
if ($#argv == 2) then
        setenv XFEHMN  $1
        setenv VER_DIR  $2
        @ paths = 1

# or If the PATHS file exists set the executable and directory paths
else if (-f PATHS) then
        source PATHS
        @ paths = 2

else
        echo 'PATHS to verification executables must be set.'
        echo 'They can be entered as command line arguments or put in a file'
        echo 'called PATHS in the current directory.'
        echo '  '
        echo 'usage: FEHM_VVSECT [xfehmn_path verification_dir]'
        exit 1
endif

@ flag = 0
# Verify that XFEHMN is executable
if (! -x $XFEHMN) then
        echo $XFEHMN' does not exist or is not an executable file.'
        @ flag = 1
endif
# Verify that VER_DIR is a directory
if (! -d $VER_DIR) then
        echo $VER_DIR' is not a directory file.'
        @ flag = 1
endif
```

FEHM_VVSECT script (continued)

```
if ($flag != 0) then
        if ($paths == 2) then
                echo 'Check your PATHS file'
        else
                echo 'Check your command line input'
        endif
        exit 1
endif

# Execute FEHMN with no input to determine version being tested
touch fehmn.files
$XFEHMN >& /dev/null; wait
set version=`cat fehmn.chk | nawk '{print $1"  "$2}'`
rm fehmn.*

echo $version': Verification started '`date` >& $results
echo '  ' >>& $results

# Execute FEHMN verification problems
if (-e $VER_DIR/RUN_VERIFICATION) then
        RUN_VERIFICATION >>& $results &; wait
        @ return = $status
        if ($return != 0) then
                echo 'Status: '$return' RUN_VERIFICATION exited with ERROR'
        endif
        echo '  ' >>& $results
else
        echo "Can't find RUN_VERIFICATION - check your verification directory"
        exit 1
endif

# Run problem comparisons and summarize results
if (-e $VER_DIR/COMPARE_RESULTS) then
        COMPARE_RESULTS -i $date >>& $results &; wait
        @ return = $status
        if ($return != 0) then
                echo 'Status: '$return' COMPARE_RESULTS exited with ERROR'
        endif
        echo '  ' >>& $results
else
        echo "Can't find COMPARE_RESULTS - check your verification directory"
        exit 1
endif

echo $version': Verification completed '`date` >>& $results
```

F. RUN_VERIFICATION Script

RUN_VERIFICATION script

```
#! /bin/csh
#       RUN_VERIFICATION
#       Script for execution of verification problems

echo 'Verification Runs for the FEHMN Application'
date
echo '  '

@ paths = 0

# Determine if executable and verification directory are defined
if (!($?XFEHMN) || !($?VER_DIR)) then

# Look for paths on the command line
        if ($#argv > 0 && $1 == "-p") then
                @ paths = 1
                shift argv
                setenv XFEHMN  $1
                shift argv
                setenv VER_DIR  $1
                shift argv

# Look in PATHS file
        else if (-f PATHS) then
                @ paths = 2
                source PATHS

        else
                echo 'PATHs to the FEHMN executable and verification directory must be set.'
                echo 'They can be entered as command line arguments or put in a file'
                echo 'called PATHS in the current directory.'
                echo '  '
                echo 'usage:RUN_VERIFICATION [-p xfehmn_path verification_dir] [problem1 problem2 ...]'
                exit 1
        endif
endif

@ flag = 0

# Verify that XFEHMN is executable
if (!(-x $XFEHMN)) then
        echo $XFEHMN' does not exist or is not an executable file.'
        @ flag = 1
endif
# Verify that VER_DIR is a directory
if (!(-d $VER_DIR)) then
        echo $VER_DIR' is not a directory.'
        @ flag = 1
endif
```

RUN_VERIFICATION script (continued)

```
if ($flag != 0) then
        if ($paths == 2) then
                echo ‘Check your PATHS file’
        else if ($paths == 1) then
                echo ‘Check your command line input’
        else
                echo ‘Check FEHM_VVSECT input’
        endif
        exit 1
endif

# If no problems are specified execute all tests
if ($#argv == 0) then
        foreach problem (avdonin dissolution doe5a dryout dual \
                fracture_transport heat2d heat3d henrys_law \
                infiltration multi_solute ramey sorption theis toronyi \
                transport3D vapor_extraction)
                echo ‘************ BEGIN ’$problem‘ **************’
                if (-e $VER_DIR/$problem && -d $problem) then
                        setenv problem $problem
                        echo ‘cd ’$problem‘; ’$VER_DIR‘/’$problem‘/run’$problem‘; wait’
                        cd $problem; $VER_DIR/$problem/run$problem; wait
                        cd ..
                else
                        echo $VER_DIR‘/’$problem‘ or ’$problem‘ does not exist’
                endif
                echo ‘************  END ’$problem‘ **************’
                echo ‘  ’
        end

# Else execute just the specified problems
else
        while ($#argv > 0)
                echo ‘************ BEGIN ’$1‘ **************’
                if (-e $VER_DIR/$1 && -d $1) then
                        setenv problem $1
                        echo ‘cd ’$1‘; ’$VER_DIR‘/’$1‘/run’$1‘; wait’
                        cd $1; $VER_DIR/$1/run$1; wait; cd ..
                else
                        echo $VER_DIR‘/’$1‘ or ’$1‘ does not exist’
                endif
                echo ‘************  END ’$1‘ **************’
                echo ‘  ’
                shift argv
        end
endif

echo ‘Verification Runs Completed’
date
```

G. COMPARE_RESULTS Script

COMPARE_RESULTS script

```
#! /bin/csh
#       COMPARE_RESULTS
#       Script for generating comparison of results and summary

echo 'Compare Results for the FEHMN Application Verification Runs'
date
echo '  '

@ path_flag = 0

# Determine if verification directory and executables are defined

if (!($?VER_DIR)) then

# Look for PATHS on the command line
        if ($#argv >= 2 && $1 == "-p") then
                @ path_flag = 1
                shift argv
                setenv VER_DIR     $1
                shift argv

# or Look in PATHS file
        else if (-f PATHS) then
                @ path_flag = 2
                source PATHS

        else

                echo 'PATHS to the verification directory must be set.'
                echo 'It can be entered as a command line argument or put in a file'
                echo 'called PATHS in the current directory.'
                echo '  '
                echo 'usage:COMPARE_RESULTS [-p verification_dir] [-i date_id] [problem1 problem2 ...]'
                exit 1

        endif

endif

# Verify that VER_DIR is a directory
if (!(-d $VER_DIR)) then
        echo $VER_DIR' is not a directory.'

         if ($path_flag == 2) then
                echo 'Check your PATHS file'
        else ($path_flag == 1) then
                echo 'Check your command line input'
```

DRAFT 3/97

COMPARE_RESULTS script (continued)

```
        else
                echo 'Check FEHM_VVSECT input'
        endif

        exit 1

endif

# Define verification executables
setenv COMPARE          $VER_DIR/COMPARE
setenv COMPARET         $VER_DIR/COMPARET
setenv SUMMARIZE        $VER_DIR/SUMMARIZE

@ flag = 0

# Verify that verification executables exist / can be executed
if (! -x $COMPARE) then
        echo $COMPARE' does not exist or is not an executable file.'
        @ flag = 1
endif
if (! -x $COMPARET) then
        echo $COMPARET' does not exist or is not an executable file.'
        @ flag = 1
endif
if (! -x $SUMMARIZE) then
        echo $SUMMARIZE' does not exist or is not an executable file.'
        @ flag = 1
endif

if ($flag != 0) then
        echo 'Check your verification directory: '$VER_DIR
        exit 1
endif

if ($#argv == 0 || $1 != "-i") then
        set date=`date +%y%m%d`
else if ($1 == "-i") then
        shift argv
        set date = $1
        shift argv
endif

set summary = SUMMARY_RPT.$date
if (-e $summary) then
        mv $summary $summary.old
endif

echo 'SUMMARY of FEHM COMPARISON TESTS '$date > $summary
echo '  ' >> $summary
```

COMPARE_RESULTS script (continued)

```
# If no problems are specified execute all tests
if ($#argv == 0) then

        foreach problem (avdonin dissolution doe5a dryout dual \
                fracture_transport heat2d heat3d henrys_law \
                infiltration multi_solute ramey sorption theis \
                toronyi transport3D vapor_extraction)
                echo '************ BEGIN '$problem' *************'
                echo '************ BEGIN '$problem' *************' >> $summary
                if (-e $VER_DIR/$problem && -d $problem) then
                        setenv problem $problem
                        echo 'cd '$problem'; comp'$problem'  '$date
                        cd $problem; $VER_DIR/$problem/comp$problem $date; cd ..
                        cat $problem/summary.$date >> $summary
                else
                        echo $VER_DIR'/'$problem' or '$problem' does not exist'
                endif
                echo '************  END '$problem' *************'
                echo '************  END '$problem' *************' >> $summary
                echo '  '
                echo '  ' >> $summary
        end

# Else compare just the specified problems
else

        while ($#argv > 0)
                echo '************ BEGIN '$1' *************'
                echo '************ BEGIN '$1' *************' >> $summary
                if (-e $VER_DIR/$1 && -d $1) then
                        setenv problem $1
                        echo 'cd '$1'; comp'$1'  '$date
                        cd $1; $VER_DIR/$1/comp$1 $date; cd ..
                        cat $1/summary.$date >> $summary
                else
                        echo $VER_DIR'/'$1' or '$1' does not exist'
                endif
                echo '************  END '$1' *************'
                echo '************  END '$1' *************' >> $summary
                echo '  '
                echo '  ' >> $summary
                shift argv
        end

endif

echo 'End Compare Results for the FEHMN Application Verification Runs'
date
```

H. Example of Problem Execution Script

runavdonin script

```
#! /bin/csh
#       runavdonin

if (! -d input) then
        set INPUT = $VER_DIR/$problem/input
        rm -f input
        ln -s $INPUT input
endif

foreach geom (84 400 800)
        echo ‘sed s/base/’$geom‘/ input/avdonin.files > fehmn.files’
        sed s/base/$geom/ input/avdonin.files > fehmn.files
        echo ‘nice ’$XFEHMN‘ &; wait’
        nice $XFEHMN &; wait
end
rm fehmn*
```

I. Example of Run Comparison Script

compavdonin script

```
#! /bin/csh
#       Comparisons for avdonin problem

if ($#argv == 0) then
        set ID = `date +%y%m%d`
else
        set ID = $1
endif

if (-e summary.$ID) then
        mv summary.$ID summary.$ID.old
endif
echo 'Summary file named: summary.'$ID

if (! -d input) then
        set INPUT = $VER_DIR/$problem/input
        rm -f input
        ln -s $INPUT input
endif

foreach type (history contour)

        foreach geom (84 400 800)
                echo 'compare '$geom'  '$type
                sed s/base/$geom/ input/avdonin.comparein.$type > comparein
                nice $COMPARE &; wait
        end

        if ($type == history) then
                sed s/param/time/ input/avdonin.summary > summarize
        else if ($type == contour) then
                sed s/param/pos/ input/avdonin.summary > summarize
        endif
        nice $SUMMARIZE >> summary.$ID; wait

end
rm comparein* summarize
```

J. Execution Log

Example of execution log

```
FEHM 01.00: Verification started Tue Jul 23 13:17:44 MDT 1996

Verification Runs for the FEHMN Application
Tue Jul 23 13:17:45 MDT 1996

************ BEGIN avdonin *************
cd avdonin; /home/fehm/verification/avdonin/runavdonin; wait
sed s/base/84/ input/avdonin.files > fehmn.files
nice /home/fehm/bin/xfehmn.95-05-01p-sun4 &; wait
[1] 25919
[1]    Done                  /home/fehm/bin/xfehmn.95-05-01p-sun4
sed s/base/400/ input/avdonin.files > fehmn.files
nice /home/fehm/bin/xfehmn.95-05-01p-sun4 &; wait
[1] 25923
[1]    Done                  /home/fehm/bin/xfehmn.95-05-01p-sun4
sed s/base/800/ input/avdonin.files > fehmn.files
nice /home/fehm/bin/xfehmn.95-05-01p-sun4 &; wait
[1] 25929
[1]    Done                  /home/fehm/bin/xfehmn.95-05-01p-sun4
************  END  avdonin *************
•
•
•
************ BEGIN vapor_extraction *************
cd vapor_extraction; /home/fehm/verification/vapor_extraction/runvapor_extraction; wait
sed s/case/iso/ input/vapextract.files > fehmn.files
nice /home/fehm/bin/xfehmn.95-05-01p-sun4 &; wait
[1] 26145
[1]    Done                  /home/fehm/bin/xfehmn.95-05-01p-sun4
sed s/case/aniso/ input/vapextract.files > fehmn.files
nice /home/fehm/bin/xfehmn.95-05-01p-sun4 &; wait
[1] 26147
[1]    Done                  /home/fehm/bin/xfehmn.95-05-01p-sun4
************  END  vapor_extraction *************

Verification Runs Completed
Tue Jul 23 20:53:53 MDT 1996
```

Example of execution log (continued)

```
Compare Results for the FEHMN Application Verification Runs
Tue Jul 23 20:53:54 MDT 1996

************ BEGIN avdonin *************
cd avdonin; compavdonin 950803
Summary file named: summary.950803
compare 84 history
[1] 26156
[1]    Done                  /home/fehm/verification/COMPARE
compare 400 history
[1] 26158
[1]    Done                  /home/fehm/verification/COMPARE
compare 800 history
[1] 26160
[1]    Done                  /home/fehm/verification/COMPARE
compare 84 contour
[1] 26164
[1]    Done                  /home/fehm/verification/COMPARE
compare 400 contour
[1] 26166
[1]    Done                  /home/fehm/verification/COMPARE
compare 800 contour
[1] 26168
[1]    Done                  /home/fehm/verification/COMPARE
************  END  avdonin *************
•
•
•
************ BEGIN vapor_extraction *************
cd vapor_extraction; compvapor_extraction 950803
Summary file named: summary.950803
compare iso contour
[1] 26587
[1]    Done                  /home/fehm/verification/COMPARE
compare aniso contour
[1] 26590
[1]    Done                  /home/fehm/verification/COMPARE
************  END  vapor_extraction *************

End Compare Results for the FEHMN Application Verification Runs
Tue Jul 23 20:56:34 MDT 1996

FEHM 01.00: Verification completed Tue Jul 23 20:56:34 MDT 1996
```

K. Summary Report

Example of summary report

SUMMARY of FEHM COMPARISON TESTS 960723

********** BEGIN avdonin **********
Avdonin Radial Heat and Mass Transfer Problem Comparison of Model and Analytical Solution -- Temperature vs Time
At R coordinate 37.5000

Test Case	Maximum Error	Maximum % Error	RMS Error
84 nodes	1.262	0.7776	0.2169E-03
400 nodes	0.4060	0.2487	0.6973E-04
800 nodes	0.3899	0.2384	0.6742E-04

Avdonin Radial Heat and Mass Transfer Problem Comparison of Model and Analytical Solution -- Temperature vs Position
At Time 0.100000E+10

Test Case	Maximum Error	Maximum % Error	RMS Error
84 nodes	0.5230	0.3237	0.1745E-03
400 nodes	0.2815	0.1744	0.3416E-04
800 nodes	0.2815	0.1744	0.2213E-04

********** END avdonin **********

•
•
•

********** BEGIN vapor_extraction **********
Vapor Extraction - Vapor Pressure vs Position

Test Case	Maximum Error	Maximum % Error	RMS Error
anisotropic	0.3066E-02	3.311	0.1436E-0
isotropic	0.1983E-02	2.195	0.8838E-04

********** END vapor_extraction **********